University of Bahrain
College of IT
Department of Computer Science
ITCS341 – Object-Oriented Systems
First Test - B

DRAGON

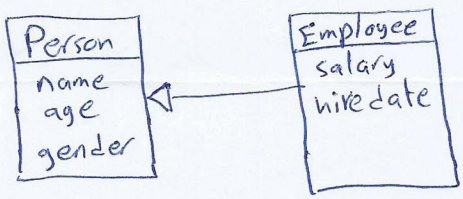
Students Name: _____ ID: _____ Section:

1. Briefly explain with an example what is meant by each of the following terms?

   a. Inheritance
   b. Object Identity
   c. Multiplicity
   d. Qualifier

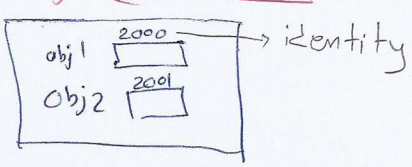
a) Inheritance is a way of code sharing of attributes and behaviors between the same application or across multiple application.

example:

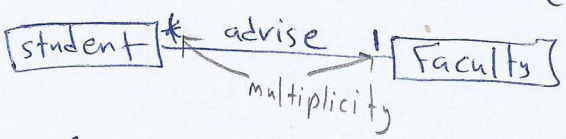


class Employee inherite the attributes of class person

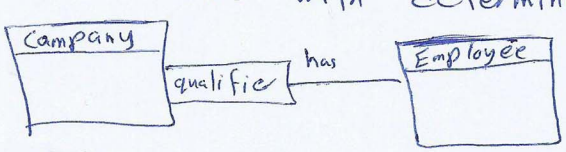
b) Object Identity is a unique link to the object which the object can be identify. ~~Identities may change in~~ different runtimes



c) Multiplicity is the number of instences of a class that can be associated with another class.



d) Qualifier helps with determining the association between classes.



University of Bahrain
College of IT
Department of Computer Science
ITCS341 – Object-Oriented Systems
First Test - B

udents Name: _____ ID:_____ Section: ___

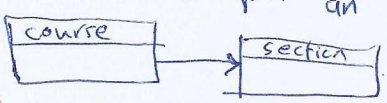
2. What is the difference between each of the following type of relationships?

   a. Association and Aggregation
   b. Static and Dynamic Aggregation
   c. Uni-Directional and Bi-Directional Associations

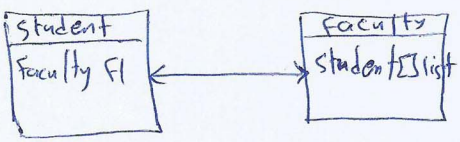
a.) association is a description of group of links with common structure and common semantics.
while aggregation is a special type of association which is a part-whole relationship.

b) Static aggregation is when an object doesn't change it's state in runtime. for example, person's ~~name~~ parents should not change.
Dynamic aggregation is when an object change it's state in run time. for example, a person has a list of children, and the list may update if the person has another child.

c) Uni-Directional association is an association in only one direction, so when we represent it, we put an instance of the other class in only one class.



Bi-Directional association is an association which work in both directions, we represent this with puting an instance of a class in both classes.



University of Bahrain
College of IT
Department of Computer Science
ITCS341 – Object-Oriented Systems
First Test - B

Students Name: _____ ID:_____ Section: ___

3. In a simple library system an item has a unique code, title, type, and library location. A library user has an ID, name, and contact number and can borrow up-to 10 items. An item can only be borrowed by a single user. The system must provide information about the items borrowed by the user as well as the user details borrowing a specific item.

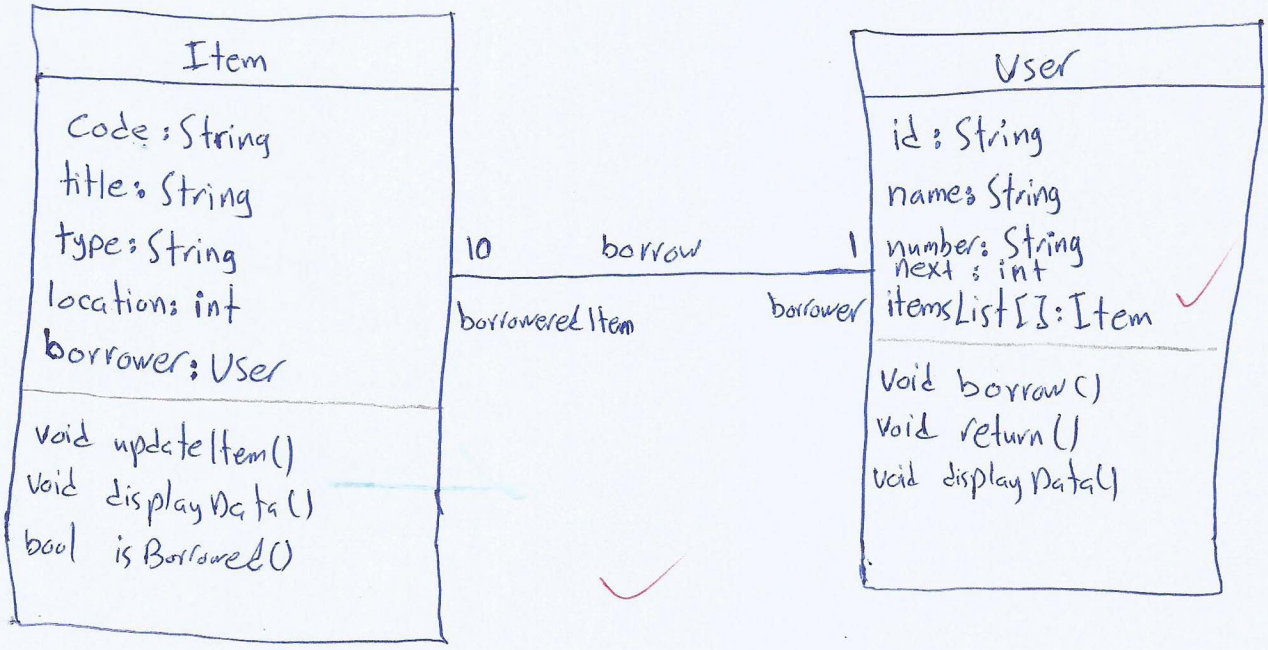
   You are required to carry out the following tasks:

   a. Using UML design the necessary classes showing their attributes, operations, and relationships (include multiplicity and end names).

   b. Show the necessary Java code for the classes including the necessary attributes and important operations for your solution (signatures only) to implement the relationship between them.

   c. Using your data structure in **b** briefly state (using pseudo code) on how you would implement the operations to borrow, return, list the items borrowed by a user, and by whom an item is borrowed.

a)



University of Bahrain
College of IT
Department of Computer Science
ITCS341 – Object-Oriented Systems
First Test - B

Students Name: _____ ID:_____ Section: ___

b)

```
class Item {

    private String code;
    private String title;
    private String type;
    private int location;
    private User borrower;

    public Item (User b);
    Public void displayData();
    public bool isBorrowed();
}
```

```
class User{
    private String id;
    private String name;
    private String number;
    private Item itemsList[];

    public user();
    public void displayData();
    public void borrow(Item t);
    public void return(Item t);
}
```

University of Bahrain
College of IT
Department of Computer Science
ITCS341 – Object-Oriented Systems
First Test - B

Students Name: _____ ID: _____ Section: ___

c)

- borrow :

```
void borrow (Item t) {
{ // if the item is not already borrowed, then we put it in the list.
   itemsList[next] = t;
   next++;
}
```

- return : search for the item in the list, and if we find it we deleted from the array and we make the user in Item = NULL.

```
Void return (Item t) {

   for (int i=0; i<next; i++) {
      if (t.getCode == itemsList[i].getCode) {
         itemsList[i].setUser = NULL;
      } itemsList[i] → remove
   }
}
```

- list of items : we go throught all items in the list and then display the data.

```
for (int i=0; i<next; i++) {
   itemsList[i].displayData();
}
```

- by whom an item is borrowed : we get this from Item class.

```
print ( item.getBorrower());
```